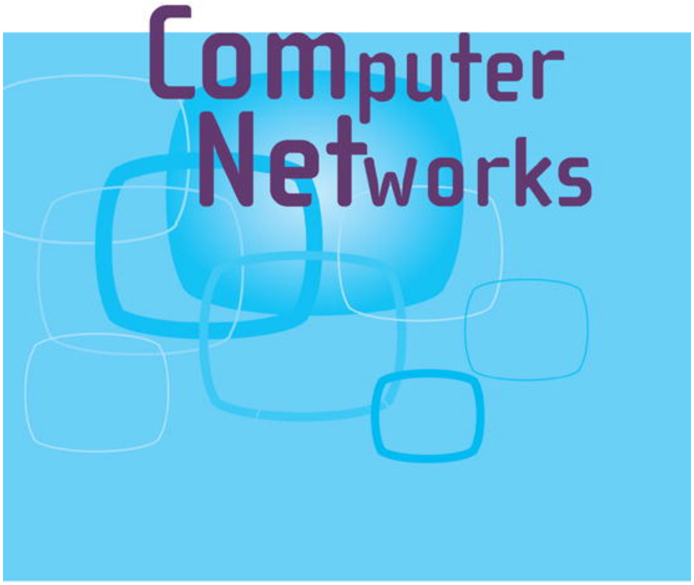
Volume 51 Issue 13 12 September 2007 ISSN 1389-1286

ELSEVIER



Available online at www.sciencedirect.com

ScienceDirect

ELSEVIER

Computer Networks 51 (2007) 3828–3843

Computer Networks

www.elsevier.com/locate/comnet

On the analysis of overlay failure detection and recovery

Zhi Li [a], Lihua Yuan [b,*], Prasant Mohapatra [c], Chen-Nee Chuah [b]

[a] *Network Systems Engineering, AT&T, United States*
[b] *Electrical and Computer Engineering, University of California, Davis, 2064 Kemper Hall, CA 95616-5294, United States*
[c] *Computer Science, University of California, Davis, CA, United States*

Received 23 October 2006; received in revised form 22 March 2007; accepted 5 April 2007
Available online 25 April 2007

Responsible Editor: I.F. Akyilidiz

Abstract

Application-layer overlay networks have been proposed as an alternative method to overcome IP-layer path anomalies and provide users with improved routing services. Running at the application layer, overlay networks usually rely on probing mechanisms for IP-path performance monitoring and failure detection. Their service performance is jointly determined by their topology, parameters of probing mechanism and failure restoration methods. In this paper, we first define metrics to evaluate the performance of overlay networks in terms of failure detection and recovery, network stability and overhead. Second, we model the overlay-based failure detection and recovery process. Through extensive simulations, we investigate how different IP-layer path failure characteristics and overlay topologies, detection and restoration parameters affect service performance of overlay networks. In particular, we examine the tradeoffs among different overlay performance metrics and the optimal performance conditions. Our study helps to understand overlay-based failure recovery and provides practical guidance to overlay network designers and administrators.
Published by Elsevier B.V.

Keywords: Overlay networks; Failure recovery; Recovery performance; Overlay design

1. Introduction

The Internet suffers from various failures and anomalies, such as optical fiber cuts [1], malicious attacks, and BGP misconfigurations [2]. Previous measurement results show that a significant amount of routing pathologies prevent pairs of hosts from connectivity 1.5% to 3.3% of the time [3], and path availabilities range from 99.6% for servers to 94.4% for broadband hosts [4]. Some routing path failures, such as intra-domain routing failures, can recover within milliseconds or seconds. However, inter-domain routing anomalies e.g., BGP failures, may take up to 30 min to recover [3]. Physical-layer failures, e.g., fiber cuts, may require days or even weeks to recover.

To provide reliable services to the application layer, each lower layer of the protocol stack, such

[*] Corresponding author. Tel.: +1 530 754 5385; fax: +1 530 752 8428.

E-mail addresses: lizhi@cs.ucdavis.edu (Z. Li), lyuan@ece.ucdavis.edu (L. Yuan), prasant@cs.ucdavis.edu (P. Mohapatra), chuah@ece.ucdavis.edu (C.-N. Chuah).

1389-1286/$ - see front matter Published by Elsevier B.V.
doi:10.1016/j.comnet.2007.04.007

as the physical layer (optical network) and the MPLS/IP layer, has its own failure detection, recovery and fault tolerance mechanism. When a failure happens, multiple protocol layers may detect it and independently adopt their own mechanisms to bypass or recover from the failure. In general, lower-layer mechanisms react faster but higher-layer mechanisms offer a finer recovery service in meeting the requirements of different users and applications. All these failure resiliency mechanisms aims to reduce user-perceived end-to-end path failures.

Recently, application-layer overlay networks [5] have been proposed to quickly detect and recover from lower-layer anomalies and improve user-perceived network resilience. Running at application layer, overlay nodes can receive and forward traffic for other nodes. Overlay nodes can send probing packets to each other to measure the performance of underlying path and detect the failure of underlying networks. When a lower-layer failure is detected, overlay nodes can re-route traffic via an alternative overlay-layer path (through other overlay nodes) and circumvent the failure. Experiments show that overlay networks can effectively overcome lower-layer anomalies and improve user-perceived network resilience [5,6].

Past research on overlay network can be broadly classified into two broad categories – (1) application overlays e.g., CAN [7], Chord [8] and Gnutella [9], which are formed by dynamic end-nodes joining and leaving the network at will, and (2) infrastructure overlays like RON [5], which has dedicated nodes committed over a prolonged period of time. Consequently, node "failures", which are actually manifestation of node membership changes, dominate the observed failures in applications overlays. In contrast, failures in infrastructure overlays are dominated by real link failures, routing anomalies and manifestation of transient failures like congestion. This paper focuses on infrastructure overlays for their capabilities to offer failure detection and recovery as a service to end-users. In addition, large-scale testbeds like PlanetLab [10] have recruited many dedicated server-class nodes around the world, making building large-scale infrastructure overlay network a realistic goal to many. In Section 2, we will discuss in detail the system model and failure recovery mechanisms used by infrastructure overlays.

Although there are a lot of work on overlay networks, not much has been done on characterizing the overlay network failure detection and recovery mechanism. This paper aims to fill this void through the following. First, we propose a comprehensive set of metrics, including reduction of failure duration, system overhead and network stability, for comparing the performance of overlay networks in terms of improving network resiliency. Second, infrastructure overlay networks have several key tunable parameters in the overlay failure detection and recovery process. For example, the frequency of probing presents a fundamental tradeoff between network overhead and the time required to detect and recover a failure. A lower probing frequency incurs less overhead but might require a longer time to detect and recover a failure. A larger probing frequency incurs more overhead but might allow the overlay network to be more responsive to the failures. Making things more complicated is the potential interaction between overlay failure recovery mechanism and the failure characteristics at lower layers and their own recovery mechanisms. For example, traffic congestion and transient network failures could produce many unnecessary overlay failure recovery events and may lead to network instability. We present a mathematical model to study the tradeoffs involving these tunable parameters and captures these complicated interactions. Third, compared to physical or IP networks, overlay networks have great flexibility in choosing their topologies since their links are logical. Our model can be extended to study the impact of topologies on the performance of failure recovery as well. The major contributions of this paper are:

- We establish a set of meaningful metrics to measure the performance of overlay failure detection and recovery and present a mathematical model to analyze them.
- We show that while it is meaningful to build a highly connected overlay for failure detection, it is counterproductive to use it for failure recovery. Based on this observation, we propose to use a different topology with smaller node degree for failure recovery.
- We study how the settings of different parameters would affect the performance of an overlay network. Based on this study, we provide meaningful advice to overlay network designers and administrators.

The rest of the paper is organized as follows. We summarize the related work in Section 5. In Section 2, we review the overlay failure detection and recovery mechanisms and define metrics to evaluate overlay network service performance. In Section 3, we characterize and analyze the failure detection and recovery mechanisms. We present simulation studies and performance analysis in Section 4 and conclude in Section 6.

2. System model

The notations used in the paper are summarized in Table 1.

2.1. Overlay link failure detection method

Earlier study on a tier-one ISP shows that single-link failures dominates all failures observed [1]. In this paper, we assume that IP-link failures are independent of each other and they are uniformly distributed over all IP-layer links. When an IP-layer link fails, a limited ratio (Q_{If}) of paths that pass through the failed IP-layer link will experience forwarding disruption, of which the duration is determined by the IP-layer failure restoration mechanisms, timer setup and topologies [1]. This duration of forwarding disruption is denoted as t_{f}. The distribution of IP-layer path failure duration is modeled as $P_{\mathrm{I}}(t_{\mathrm{f}})$.

Overlay networks, such as RON [5] and Akamai [6], usually adopt the following method to detect overlay link (IP-layer path) anomalies and perform overlay-layer failure recovery. Each overlay node periodically sends probing packets, which includes the performance of its adjacent overlay links to other overlay nodes every probing interval (T_{p} seconds). Upon receiving a probing packet, an overlay node replies to the sender with an ACK packet. If an ACK packet is not received within a predefined time (RTT or longer), the packet sender will deem the probing packet has timed out and continuously send additional k probing packets at interval T_t (fast retransmission interval). If all the k packets time out, the source node will consider this overlay link has failed. Thus, the source node employs application-layer path recovery mechanism (finding an alternative overlay path) based on its knowledge of overlay network link state information. We define the time gap between detecting a failure and finding an overlay path to overcome the failure as Overlay Failure Recovery Time (T_{rt}), which is mainly composed of a hold-off timer to avoid the race condition between IP layer and overlays. After an overlay link is detected down, overlay nodes will continue sending probing packets every T_{p} seconds to check if the connectivity is back.

The value of T_t is usually fixed to a value higher than the maximal expected round trip delay and correlated packet loss interval to avoid transient congestion caused consecutive loss [11]. As the correlated packet loss interval in the Internet is usually around one second, similar to the TCP SYN timeout value, T_t is usually between 1 and 3 s. For the rest of our discussion, we assume the value of T_t is 3 s. Same as RON [5], we set k at three. The probing interval (T_{p}) and the average number of neighbors determine the probing overhead of each overlay node. With a large value of T_{p}, overlay nodes cannot quickly detect IP-path failures, which will result in a large amount of packet loss. However, a small T_{p} will result in higher probing and routing overhead as well as path instability.

2.2. Other failure detection algorithms

There are some previous work using more complicated failure detection schemes to achieve faster failure detection. We broadly classify these schemes into three categories.

- *Peer information sharing* schemes [11,12] propose for nodes to share failure information. However, these algorithms are proposed in the context of

Table 1
Notations for analysis

Notion	Explanation	Notion	Explanation
t_{f}	IP path failure duration (duration of forwarding disruption)	T_{p}	overlay network probing interval
T_t	Fast retransmission time interval	T_{rt}	overlay failure recovery time
$P_{\mathrm{I}}(t_{\mathrm{f}})$	probability density function (Pdf) of path failure with duration t_{f} based on IP recovery	$P_{\mathrm{O}}(t_{\mathrm{f}})$	Pdf of path failure with duration t_{f} with overlay recovery
$P_{\mathrm{Od}}(t)$	Pdf of an IP-path failure detected by the overlay t time after it happens	$Q_{\mathrm{Od}}(t_{\mathrm{f}})$	Overlay failure detection ratio for IP-path failures with duration t_{f}
Q_{If}	ratio of affected IP paths caused by an IP-layer link failure	Q_{Or}	overlay failure recovery ratio (ratio of finding alternate paths)
$\overline{Q_{\mathrm{Od}}}$	average IP failure detection ratio at the overlay layer	Q_{Orl}	failure recovery ratio loss compared to best performance

application overlay networks in which failures are dominant by nodes joining and leaving the network. Sharing positive or negative (peer up or down) information [11] could be beneficial for failure detection since other nodes should observe a node failure consistently regardless of where the observing overlay node is located. In contrast, an infrastructure overlay with dedicated server-class nodes experience significantly less failures than link failures. Two overlay links may or may not share the failed lower-layer link. Without corresponding lower-layer topology information, the knowledge about the failure of one overlay link cannot help to infer status of other overlay links.

- *Application information sharing* schemes propose to use application packets as probing packets and the corresponding ACKs or NACKs [13] to monitor the link. Such scheme requires integration of overlay network protocol and specific applications and its performance depends heavily on the traffic profile of the application.
- *Cross-layer information sharing* schemes propose for lower layers to provide overlay node access to their information like routing and/or notify overlay node for any failure they detected. However, such explicit support from lower layers requires modification at lower layers, which is not always feasible. Since overlay network is a user-level process, sharing lower-layer information that are normally maintained at kernel level might pose a security risk.

While each of these complex schemes has its respective advantages, this paper choose to focus on the failure detection algorithm proposed by RON for its general applicability. The failure detection algorithm we considered can be achieved by any overlay nodes as long as they can send application-layer packets. It does not require any explicit support from lower layers and relies on any specific traffic pattern of the applications. Studying this scenario can give us better understanding on tradeoffs and fundamental limits of overlay network itself.

2.3. Overlay network performance metrics

2.3.1. Average failure duration reduction

An important metric for overlay networks is to the capability to quickly detect lower-layer path failures and find alternative paths. We introduce the definition of *Average Failure Duration Reduction* (*AFDR*) to evaluate the performance in terms of bypassing IP-layer path failures and providing resilient routing services. It is defined as

$$\mathrm{AFDR} = \int_0^\infty t_\mathrm{f} P_\mathrm{I}(t_\mathrm{f})\,\mathrm{d}t - \int_0^\infty t_\mathrm{f} P_\mathrm{O}(t_\mathrm{f})\,\mathrm{d}t. \tag{1}$$

In the above equation, $P_\mathrm{O}(t_\mathrm{f})$ is the distribution of path failure duration experienced by end-users when passing through overlay networks. As a result, $\int_0^\infty t_\mathrm{f} P_\mathrm{O}(t_\mathrm{f})\,\mathrm{d}t$ is the average path failure duration (mean time to repair, MTTR) on overlay networks. In contrast, $\int_0^\infty t_\mathrm{f} P_\mathrm{I}(t_\mathrm{f})\,\mathrm{d}t$ is the MTTR directly on the IP networks.

2.3.2. Overlay service overhead

Overlay networks cannot provide service without any cost. As stated above, overlay networks can only detect failures and retrieve overlay link performance by periodically sending and receiving probing packets. As overlay networks usually provide routing service based on link state routing protocols, the total routing service overhead (O) is not only comprised of probing traffic overhead (O_P), but also the routing traffic overhead of sending and receiving overlay link state information (O_R).

2.3.3. Overlay routing service instability

When IP or lower-layer failures happen, both the IP layer and the overlay layer will perform their own recovery mechanisms to bypass the failures. If the IP layer recovers faster, overlay-layer recovery is unnecessary and results in undesirable path oscillations. In consequence, these oscillations lead to traffic delay variations, out-of-order packet delivery, and reduced end-to-end throughput. To evaluate the path stability of service paths, we use the number of user path switches per IP failure, denoted N_s, which includes the switches between IP paths and overlay paths as well as the switches just among overlay paths.

In summary, the goal of overlay networks is to reduce the average failure duration and provide stable data forwarding paths while incurring acceptable overhead.

3. Analysis of overlay failure detection and recovery

In this section, we model the failure detection and recovery process in overlay networks. Based on this model, we formulate performance metrics and investigate how different parameters, such as the value of T_p and T_{rt}, can affect different overlay networks service performance metrics.

3.1. Failure detection

Fig. 1 depicts an example of the overlay-layer failure detection process. In this figure, node $\mathscr{A}$ probes the overlay link connecting it to node $\mathscr{B}$. Node $\mathscr{A}$ sends the first probing packet at time 0 and receives an ACK message from node $\mathscr{B}$. Suppose the IP-layer path between node $\mathscr{A}$ and node $\mathscr{B}$ fails at T_o. Node $\mathscr{A}$ will not detect the failure event right away. At $T_{\rm p}$, $\mathscr{A}$ sends the second probing packet. At $T_{\rm p}+T_t$, the timer for the second probing packet has expired, and $\mathscr{A}$ then sends an additional $k-1$ probing packets. If the IP-layer path failure lasts longer than $T_{\rm p}-T_o+kT_t$, $\mathscr{A}$ infers that the overlay link has failed and tries to find an alternative overlay path to $\mathscr{B}$.

As IP-layer path failure durations are variable, if a failure (or a transient traffic congestion) is recovered before $\mathscr{A}$'s second probing packet times out, the anomaly cannot be detected by node $\mathscr{A}$ at all. The relationship between failure occurrence time (T_o after sending out the latest probing packet) and the minimal detectable failure duration is described in Fig. 2. The X-axis is the failure occurrence time while Y-axis is the minimum detectable failure duration. Y-axis value can be expressed as $f(x)=T_{\rm p}+kT_t-x$. From the figure, we can observe that if a failure occurs at time 0 after the previous probing packet, the detectable failures durations should be at least $T_{\rm p}+kT_t$. However, if a failure occurs at $T_{\rm p}$, it could still be detectable even though it only lasts for kT_t.

Given an IP-path failure lasting for duration $t_{\rm f}$, the probability that it is detected ($Q_{\rm Od}(t_{\rm f})$) can be expressed by Eq. (2). All the IP-path failures that last longer than $T_{\rm p}+kT_t$ can definitely be detected at the overlay layer while those last shorter than kT_t will not be detected at all.

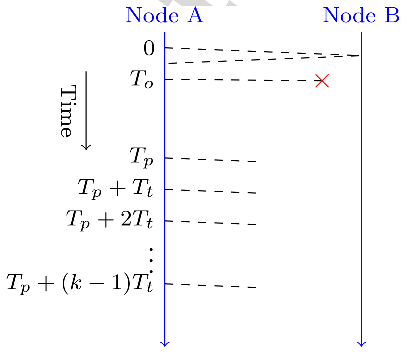


Fig. 1. Failure detection process.

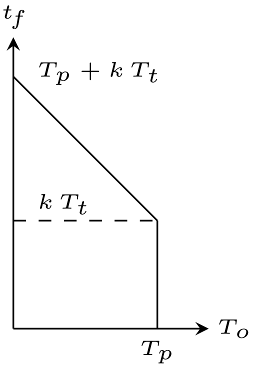


Fig. 2. Occurrence time vs. minimum detectable duration.

$$Q_{\rm Od}(t_{\rm f})=\begin{cases}0 & \text{if } t_{\rm f}<kT_t,\\ \frac{t_{\rm f}-kT_t}{T_p} & \text{if } kT_t\leqslant t_{\rm f}\leqslant T_{\rm p}+kT_t,\\ 1 & \text{if } t_{\rm f}>T_{\rm p}+kT_t.\end{cases} \tag{2}$$

The average detectable failure ratio at the overlay layer ($\overline{Q_{\rm Od}}$) can be expressed as

$$\overline{Q_{\rm Od}}=\int_0^\infty Q_{\rm Od}(t_{\rm f})\cdot P_{\rm I}(t_{\rm f})\,{\rm d}t_{\rm f}. \tag{3}$$

The dashed curve in Fig. 3(b) is an example distribution of IP-path failure duration. Based on Eq. (2) (described in Fig. 3(a)), the corresponding distribution of detected IP-path failure durations can be described by the solid curve in Fig. 3(b). The first part of the distribution curve, $\mathscr{A}$–$\mathscr{B}$, is determined by the value of $T_{\rm p}$ and kT_t. The value of kT_t determines the location of $\mathscr{A}$ while the value of $T_{\rm p}$ determines the slope of $\mathscr{A}$–$\mathscr{B}$. The smaller value of kT_t and $T_{\rm p}$ will help overlay networks detect more IP path failures.

As a failure can start at anytime (between 0 and $T_{\rm p}$) after the previous successful probe, there will be some detection delay between the failure occurrence time and detection time. The distribution of failures with respect to the detection delay is described in Eq. (4), which reflects the minimal possible recovery delay via overlays.

$$P_{\rm Od}(t_{\rm f})=\begin{cases}\int_{t_{\rm f}}^{\infty}\frac{P_{\rm I}(x)Q_{\rm Od}(x)}{T_{\rm p}}\,{\rm d}x & \text{if } kT_t\leqslant t_{\rm f}\leqslant T_{\rm p}+kT_t,\\ 0 & \text{else.}\end{cases} \tag{4}$$

3.2. Failure recovery

Overlay nodes initiate failure recovery and try to find alternative overlay-layer paths to bypass their detectable failures based on their knowledge of

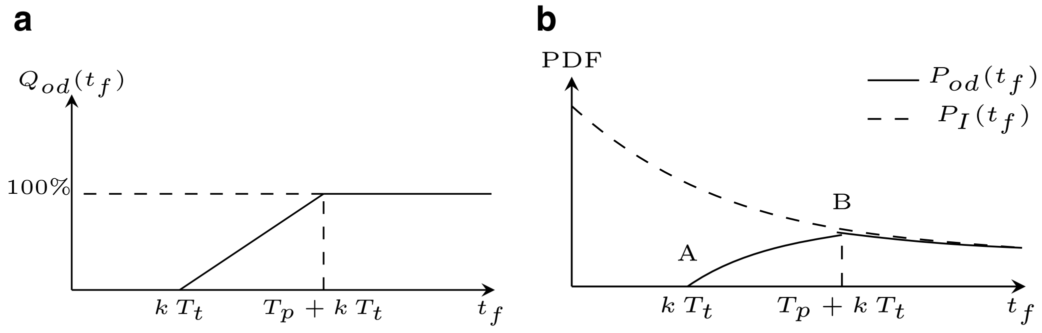


Fig. 3. Distribution of detectable IP path failure duration. (a) Detection ratio and (b) failure duration.

global overlay link state information. Similar to other link-state based routing protocols (such as OSPF), it is necessary that probing and routing state update events of each overlay node are not synchronized. However, as shown in the following analysis, this asynchronous behavior may decrease overlay service performance and path stability to some extent.

When an overlay node detects a failure on a neighboring link, it will try to re-route through other nodes to reach its destination. The search for alternative routes is based on the *local* information about of other overlay links. Since overlay nodes do not synchronize their probing activities, some nodes will require more time to detect overlay link failures than other nodes, even though overlay link failures caused by the same IP-layer failure event happen at the same time in reality. Such asynchronized detection may cause nodes to have incorrect link state information, resulting in sub-optimal route decision and negatively affect the performance of failure recovery and network stability.

In the following, we consider a simple overlay topology (Fig. 4) and use it to illustrate our discussions on the impact of such asynchronous failure detection on the performance of overlay failure recovery. To simplify the analysis, we ignore the path propagation delay between overlay nodes.

Fig. 5 depicts a typical scenario of such asynchronous failure detection. Assume IP link $\mathscr{F}$–$\mathscr{B}$ fails at time t and overlay node $\mathscr{A}$ detects the resulted failure of overlay link $\mathscr{A}$–$\mathscr{B}$ at time $T_{\rm p}+kT_t$ ($0 \leqslant t \leqslant T_{\rm p}$). We denote T_{rt} the time for $\mathscr{A}$ to find an alternative path after detecting the failure. Another overlay link $\mathscr{C}$–$\mathscr{B}$ might also fail because it passes through $\mathscr{F}$–$\mathscr{B}$. If $\mathscr{A}$ does not have correct information about $\mathscr{C}$–$\mathscr{B}$ at time $T_{\rm p}+kT_t+T_{rt}$, it may make suboptimal decision for failure recovery. Although overlay link failures that are caused by the same IP failure start at the same time, their failure durations could be different since they are determined by the underlying IP-layer failure recovery mechanisms,. If $\mathscr{C}$–$\mathscr{B}$ remains failed at time $T_{\rm p}+kT_t+T_{rt}$, i.e., the failure duration of $\mathscr{C}$–$\mathscr{B}$ (denoted as $t_{\rm f}$) is longer than $T_{\rm p}-t+kT_t+T_{rt}$, $\mathscr{A}$ should ideally be notified in time and avoid using $\mathscr{C}$–$\mathscr{B}$ for failure recovery. Note that the probability that the failure duration of $\mathscr{C}$–$\mathscr{B}$ is larger than $T_{\rm p}-t+kT_t+T_{rt}$ can be found as $\int_{T_{\rm p}+kT_t+T_{rt}-t}^{\infty} P_{\rm I}(x)\,{\rm d}x$.

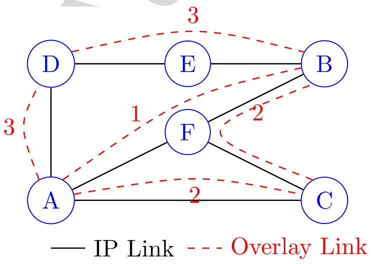


Fig. 4. Example overlay network.

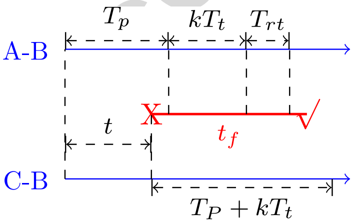


Fig. 5. Asynchronous failure detection and recovery.

Since an overlay node requires at least kT_t time to confirm a link failure, $\mathscr{C}$ can only notify $\mathscr{A}$ about the failure in time if $\mathscr{C}$ sends out one probing packet before $T_{\rm p}+T_{rt}$. Given a probing interval of $T_{\rm p}$, the first probing packet from $\mathscr{C}$ to $\mathscr{B}$ after the IP-layer failure event is uniformly distributed within the interval of $[t,t+T_{\rm p}]$. Therefore, at the time of making routing decision, the probability of $\mathscr{A}$ can correctly identify $\mathscr{C}$–$\mathscr{B}$ as failed is $\frac{T_{\rm p}+T_{rt}-t}{T_{\rm p}}$ (if $t \geqslant T_{rt}$) or 1 (if $t < T_{rt}$).

Denote Q_{fg} as the probability of false negatives – a failed overlay links is considered good for failure recovery purposes. Q_{fg} can be found as

$$
\begin{aligned}
Q_{fg}^{T_{rt}} &= \int_{T_{rt}}^{T_{\mathrm{p}}} \frac{1}{T_{\mathrm{p}}}\left(1-\frac{T_{\mathrm{p}}+T_{rt}-t}{T_{\mathrm{p}}}\right)\\
&\quad \times \int_{T_{\mathrm{p}}-t+kT_t+T_{rt}}^{\infty} P_{\mathrm{I}}(x)\,\mathrm{d}x\,\mathrm{d}t\\
&= \int_{T_{rt}}^{T_{\mathrm{p}}} \frac{t-T_{rt}}{T_{\mathrm{p}}^2}\int_{T_{\mathrm{p}}-t+kT_t+T_{rt}}^{\infty} P_{\mathrm{I}}(x)\,\mathrm{d}x\,\mathrm{d}t. \qquad (5)
\end{aligned}
$$

Note that if the failure duration of $\mathscr{C}$–$\mathscr{B}$ is shorter than $T_{\mathrm{p}}-t+kT_t+T_{rt}$, $\mathscr{A}$ will not notice the failure either. However, we do not consider this as false negatives since $\mathscr{C}$–$\mathscr{B}$ has recovered at the time $\mathscr{A}$ needs to make routing decisions. Although $\mathscr{A}$ fails to know that $\mathscr{C}$–$\mathscr{B}$ actually failed for a short period of time, the information it has at the time of making decisions is accurate.

Based on above result, the probability of having correct information about a failed link ($Q_{ff}^{T_{rt}}$), (consider the failed link $\mathscr{C}$–$\mathscr{B}$ as failed), can be found as

$$Q_{ff}^{T_{rt}} = \int_{0}^{T_{\mathrm{p}}} \frac{1}{T_{\mathrm{p}}}\int_{T_{\mathrm{p}}-t+kT_t+T_{rt}}^{\infty} P_{\mathrm{I}}(x)\,\mathrm{d}x\,\mathrm{d}t - Q_{fg}^{T_{rt}}. \qquad (6)$$

Note that Fig. 5 and our above discussions only illustrate the simplest scenario of such delayed convergence. In reality, such staggered convergence events may have cascading effects and incur long time service path instability among different overlay paths.

As overlay nodes perform failure recovery based on overlay link performance probing results, transient failures or traffic congestion may provide overlay nodes with incorrect information of overlay link performance. If failures are recovered (or congestion disappears) at the lower-layer before overlay-layer failure recovery finishes (or update messages are sent), the overlay link failures should not be counted as *failed* when a node performs overlay-layer failure recovery. Otherwise, we call them *false positives*. False positives can cause the following undesirable effects: (1) overlay nodes send out redundant performance update messages; (2) overlay nodes perform unnecessary failure recovery causing routing service oscillation; (3) overlay nodes have incorrect overlay link state information and the probability of bypassing the failed links is reduced. This probability is described in the following equation:

$$Q_{gf}^{T_{rt}} = \int_{0}^{T_{\mathrm{p}}} \frac{1}{T_{\mathrm{p}}}\int_{kT_t}^{T_{\mathrm{p}}-t+kT_t+T_{rt}} P_{\mathrm{I}}(x)Q_{\mathrm{Od}}(x)\,\mathrm{d}x\,\mathrm{d}t. \qquad (7)$$

Specifically, this is the probability of another overlay link failures that is recovered before an overlay node performs recovery activity for a failed overlay link (*g*ood links are deemed as *f*ailed links).

Based on above result, for the following ratio:

$$Q_{gg}^{T_{rt}} = \int_{0}^{T_{\mathrm{p}}} \frac{1}{T_{\mathrm{p}}}\int_{0}^{T_{\mathrm{p}}-t+kT_t+T_{rt}} P_{\mathrm{I}}(x)\,\mathrm{d}x\,\mathrm{d}t - Q_{gf}^{T_{rt}}, \qquad (8)$$

a good overlay link (such as $\mathscr{C}$–$\mathscr{B}$) will be considered as a good link (correct information) by another node (such as node $\mathscr{A}$) when it searches for an alternate overlay path.

Suppose the IP link failure ratio is Q_{If} (IP paths affected by an underlying IP-layer failure event). Then, $Q_g(=(1-Q_{If})+Q_{If}Q_{gg}^{T_{rt}})$ is the ratio of operational overlay links for which an overlay node also has the correct information of those link performance. Let L be the average number of overlay hops overlay paths pass through (or the number of intermediate overlay nodes). Compared to the ideal case, overlay nodes can fail (or incur delay) while selecting each link for failure recovery by either misidentifying a failed overlay link as a good one or vice versa. Thus, the failure recovery ratio loss (Q_{Orl}) can be derived as

$$Q_{\mathrm{Orl}} = 1-\left(\frac{Q_g}{Q_g+Q_{If}Q_{gf}^{T_{rt}}}\right)^{L} \times \left(\frac{Q_g}{Q_g+Q_{If}Q_{fg}^{T_{rt}}}\right)^{L}. \qquad (9)$$

As shown in [2,5], when $L=2$, overlay networks can achieve good performance by overcoming around 50% of IP-layer path failures. In addition, in large and well-connected IP networks like the Internet, the value of Q_{If} will not be very large. Based on these facts, we can conclude that the failure recovery ratio loss will not be very large most of the time.

Suppose the ideal (best) failure recovery ratio of an overlay network is Q_{rideal}. The value will be the failure recovery ratio if each overlay node has accurate global overlay link state information. However, as shown above, the value of T_{p}, kT_t and T_{rt} will affect the accuracy of information observed by each overlay node. In Eq. (9), we have defined Q_{Orl} as the overlay failure recovery ratio loss compared to the ideal performance. The actual failure recovery ratio can be defined as

$$Q_{\mathrm{Or}} = (1-Q_{\mathrm{Orl}})Q_{\mathrm{rireal}}. \qquad (10)$$

3.3. Impact of overlay topology

As overlay networks are built at the application layer, the administrator can determine whether to build one overlay link between any pair of overlay nodes. We can use various topologies to connect the overlay nodes, based on which the overlay traffic can be forwarded to bypass the IP path failures.

Overlay overhead is composed of two parts: path probing overhead and routing state update overhead. Based on path probing activity, each overlay node can detect its adjacent overlay link performance as well as IP-path failure events. This part of the active probing overhead is determined by the overall size of the corresponding overlay networks. We can also further reduce the overhead by passively listening to the passing overlay or IP traffic to infer the performance information. Routing state update overhead is determined by the average number of adjacent overlay nodes in the corresponding overlay topologies.

In this paper, we assume that the overlay nodes perform active probing to retrieve the overlay path performance information. Probing overhead (O_{p}) can be expressed by Eq. (11):

$$O_{\mathrm{p}} \propto \frac{S \times D}{T_{\mathrm{p}}}, \tag{11}$$

where S is the size of probing packet and D is the average number of adjacent overlay nodes in the overlay topology.

Routing overhead (O_{r}) is strongly affected by the design and goal of the routing protocol. In RON [5], in addition to link availability, updated link performance information are sent together with every probing packets. The routing update overhead can therefore be expressed as

$$O_{\mathrm{r}} \propto \frac{N \times (H + P \times D)}{T_{\mathrm{p}}}, \tag{12}$$

where H is the header size of routing packet, P contains the information describing path to each peer and N is the size of overlay networks. If the routing protocol only provides failure recovery, link state advertisement (LSA) need to be sent only when there is a change of link status (up or down). In such case, routing overhead can be significantly reduced.

As the overlay data forwarding paths are based on top of overlay topologies, it is obvious that overlay topologies determine its best failure recovery ratio (Q_{rireal}, so as to the practical failure recovery ratio, Q_{Or}).

The practical value of ideal failure recovery ratio an overlay network can provide is determined by the size of the overlay network, the underlying size of the IP network as well the topology of the IP network. To show different node degree's impact on failure recovery ratio, we investigate a 20-node overlay network on top of an 100-node IP network, which is connected by a grid topology. We vary the average node degree of the overlay network. If the node degree is less than 19 (full-mesh case), each overlay node randomly chooses overlay neighbors. The average ideal failure recovery ratio is shown in Fig. 6. The different curves show the ideal failure recovery ratios on top of different number of concurrent IP link failures. From this figure, we can observe that the ideal failure recovery ratio is not improved after the average node degree is above four or five. This means that some additional overlay links are not helpful in terms of failure recovery. On the other side, we can see the higher node degree is needed when there are large amount of concurrent IP link failure events.

The corresponding per-node overlay routing overhead is shown in Fig. 7. From the figure, we can see the overlay routing overhead varies a lot under different values of average node degree, which is even more severe if we do not include the node probing overhead (O_{p}). Considering the higher node degree does not necessarily mean higher failure recovery ratio, it is possible for us to reduce overhead without degrading overlay service performance.

3.4. Mean time to repair (MTTR)

Based on above derivation of Q_{Od} (failure detection ratio for an IP path failure with duration as t

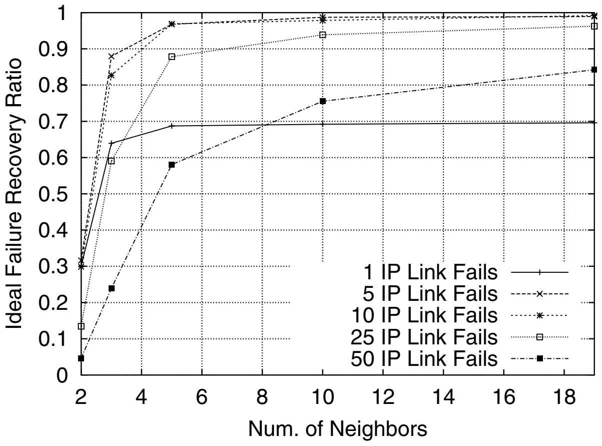


Fig. 6. Node degree vs. ideal failure recovery ratio.

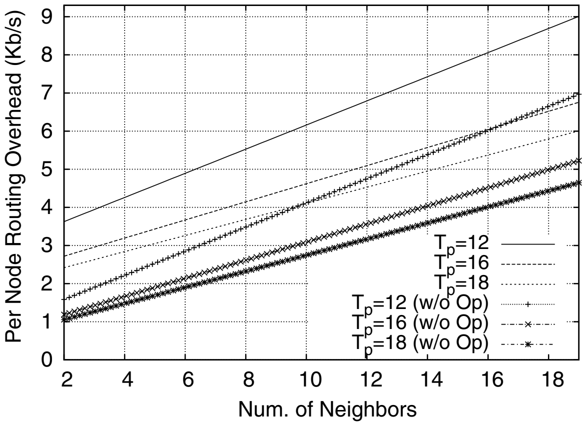


Fig. 7. Node degree vs. overhead.

second), P_{Od} (the probability of detecting a failure t seconds after it happens) and Q_{Or} (failure recovery ratio), we can obtain the analytical result of path failure duration on top of overlays ($P_{\mathrm{O}}(t)$) as defined as

$$P_{\mathrm{O}}(t_{\mathrm{f}}) = \begin{cases} P_{\mathrm{I}}(t_{\mathrm{f}}) & \text{if } t_{\mathrm{f}} < kT_t, \\ P_{\mathrm{I}}(t_{\mathrm{f}})(1 - Q_{\mathrm{Od}}(t_{\mathrm{f}})Q_{\mathrm{Or}}) & \\ +P_{\mathrm{Od}}(t_{\mathrm{f}} - T_{rt})Q_{\mathrm{Or}} & \text{if } t_{\mathrm{f}} \geqslant kT_t. \end{cases} \tag{13}$$

From Eq. (13), we can see that the overlay networks cannot provide help for failures that last shorter than kT_t and therefore cannot be detected. For failures that last longer than kT_t, with probability $P_{\mathrm{I}}(t_{\mathrm{f}})(1 - Q_{\mathrm{Od}}(t_{\mathrm{f}})Q_{\mathrm{Or}})$ they are not recoverable. With overlay providing failure recovery services, the probability of a failure lasts t_{f} is $P_{\mathrm{Od}}(t_{\mathrm{f}} - T_{rt})Q_{\mathrm{Or}}$.

3.5. Average failure duration reduction (AFDR)

Based on the above results, for an overlay link failure with a duration of t_{f} at the IP layer, the average time between failure occurrence time and failure detected time is

$$D_t(t_{\mathrm{f}}) = \begin{cases} \int_{kT_t}^{t} x\,\mathrm{d}x & \text{if } t_{\mathrm{f}} \leqslant T_{\mathrm{p}} + kT_t, \\ \int_{kT_t}^{T_{\mathrm{p}}+kT_t} x\,\mathrm{d}x & \text{if } t_{\mathrm{f}} > T_{\mathrm{p}} + kT_t. \end{cases} \tag{14}$$

AFDR can be obtained by comparing those detectable and recoverable failures' IP path failure durations and the durations on top of overlays. That is, $D_t(t)$ plus overlay failure recovery time (mainly overlay hold-off timer). The value of AFDR can be described by the following equation:

$$\int_0^\infty P_{\mathrm{I}}(t)Q_d(t)Q_{or}(t - (T_{rt} + tD_t))\,\mathrm{d}t, \tag{15}$$

where $T_{rt} + D_t t$ is the failure duration on top of overlay for detectable IP-path failure.

3.6. Path stability

Overlay networks respond to IP path failures by re-routing through other overlay nodes. In addition to extra overhead, path switches also incur end-to-end delay variations, out-of-order packet delivery and reduced throughput. Unlike the IP-layer, where single link failure dominates, an overlay network can have multiple correlated link failures caused by a single IP failure. As illustrated in Fig. 4, a failure at the IP link $\mathscr{F}$–$\mathscr{B}$ will cause two overlay link failures at $\mathscr{A}$–$\mathscr{B}$ and $\mathscr{C}$–$\mathscr{B}$.

Since the probing timers on different overlay nodes are not, and should not be, synchronized, correlated overlay link failures are detected at different times. Consequently, overlay nodes may have inconsistent and inaccurate link state information for recovery purposes. Again, consider an IP failure at link $\mathscr{F}$–$\mathscr{B}$ in Fig. 4, if $\mathscr{A}$ detects the failure of overlay link $\mathscr{A}$–$\mathscr{B}$ first, it may decide to re-route through $\mathscr{A}$–$\mathscr{C}$–$\mathscr{B}$. This path switch is however meaningless. When C detects the overlay failure on link $\mathscr{C}$–$\mathscr{B}$ later, A need to switch again to $\mathscr{A}$–$\mathscr{D}$–$\mathscr{B}$. The overlay link $\mathscr{C}$–$\mathscr{B}$, in this particular case, only incurs additional path switches and delays the re-convergence and the failure recovery process.

The durations of correlated IP-path failures might also be different since the IP layer take different time to recover different IP path. In Fig. 4, the IP layer could find path $\mathscr{A}$–$\mathscr{D}$–$\mathscr{E}$–$\mathscr{B}$ before it finds $\mathscr{C}$–$\mathscr{A}$–$\mathscr{D}$–$\mathscr{E}$–$\mathscr{B}$. Therefore, the overlay node $\mathscr{C}$, in order to reach $\mathscr{B}$, could be moving from overlay path $\mathscr{C}$–$\mathscr{A}$–$\mathscr{D}$–$\mathscr{B}$ to overlay path $\mathscr{C}$–$\mathscr{A}$–$\mathscr{B}$ to $\mathscr{C}$–$\mathscr{B}$. Eventually, $\mathscr{C}$ might return to use IP-layer connectivity between $\mathscr{C}$ and $\mathscr{B}$ since overlay-based routing is more expensive. During the entire process, the IP-layer path between $\mathscr{C}$ and $\mathscr{B}$ does not change. But the end-user will experience path oscillations.

In the following, we analyze overlay path switches based on Fig. 8. The curve is the IP-layer failure distribution without using overlay networks. The Y-axis can be seen as the probability of the failure duration taking on a corresponding X-axis value.

Suppose an IP-layer failure is detected at time t ($t \in [kT_t,\ T_{\mathrm{p}} + kT_t]$). After a short Hold-off Timer

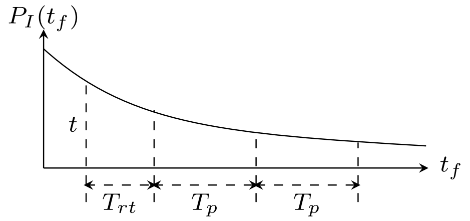


Fig. 8. Path switches and failure distribution.

(T_{rt} to avoid the race condition between two layers), if the failure is recoverable (with probability of Q_{Orl}), the traffic will be redirected to overlay paths. After this, the overlay node will continue to probe the original IP path performance every T_{p} seconds. If the IP path failure is recovered by the IP path, the traffic will be redirected to original IP path after T_{p}. The number of switches within time gap T_{p} can be expressed in the following equation:

$$2Q_{\mathrm{Orl}} \times \int_{kT_t}^{kT_t+T_{\mathrm{p}}} \frac{1}{T_{\mathrm{p}}} \int_{t+T_{rt}}^{t+T_{rt}+T_{\mathrm{p}}} P_{\mathrm{I}}(x)\,\mathrm{d}x\,\mathrm{d}t. \tag{16}$$

Similarly, at time $t + T_{rt} + 2T_{\mathrm{p}}$, some IP path failures can also be recovered and overlay traffic will be directed back to the original IP paths again. The additional number of switches with time gap $2T_{\mathrm{p}}$ can be expressed as

$$2Q_{\mathrm{Orl}} \times \int_{kT_t}^{kT_t+T_{\mathrm{p}}} \frac{1}{T_{\mathrm{p}}} \int_{t+T_{rt}+T_{\mathrm{p}}}^{t+T_{rt}+2T_{\mathrm{p}}} P_{\mathrm{I}}(x)\,\mathrm{d}x\,\mathrm{d}t. \tag{17}$$

Overall, the number of switches per overlay link failure ($N_{\mathrm{s}}^{\mathrm{o}}$) can be described in the following equation:

$$\begin{aligned} N_{\mathrm{s}}^{\mathrm{o}} &= 2Q_{\mathrm{Orl}} \int_{kT_t}^{kT_t+T_{\mathrm{p}}} \sum_{n=1}^{\infty} \frac{1}{T_{\mathrm{p}}} \\ &\quad \times \int_{t+T_{rt}+(n-1)T_{\mathrm{p}}}^{t+T_{rt}+nT_{\mathrm{p}}} P_{\mathrm{I}}(x)\,\mathrm{d}x\,\mathrm{d}t \\ &= 2Q_{\mathrm{Orl}} \int_{kT_t}^{kT_t+T_{\mathrm{p}}} \frac{1}{T_{\mathrm{p}}} \int_{t+T_{rt}}^{\infty} P_{\mathrm{I}}(x)\,\mathrm{d}x\,\mathrm{d}t. \end{aligned} \tag{18}$$

As we defined in Section 2, an IP failure could incur multiple overlay failures and the ratio of affected IP-path is Q_{If}. For a full-mesh overlay network with N nodes, the total number of affected overlay links is N^2Q_{If}. Consequently, the number of path switches per IP failure (N_{s}) is

$$N_{\mathrm{s}} = N^2 Q_{If} N_{\mathrm{s}}^{\mathrm{o}}. \tag{19}$$

4. Simulation study

In this section, we perform simulation studies to validate our analysis and study the tradeoffs among different overlay performance metrics. Unfortunately, we cannot evaluate our study on a real overlay network like PlanetLab [10] for two reasons. First, actual failures on the Internet is highly random and experiments based on this will not be repeatable. Second, injecting controlled IP-layer failures into the Internet is difficult and impractical.

4.1. Simulation setup

4.1.1. Network model

In our simulation, we use a two-level power-law topology with 1000 nodes and 4000 edges generated by BRITE [14] with the default parameters. The lower level is based on the Waxman model with parameters $\alpha = 0.15$ and $\beta = 0.2$. The higher level is based on the Barabasi-Albert model without rewiring. At the overlay layer, we randomly select 50 nodes from the 1000 IP-layer nodes as overlay nodes and construct the overlay network. We use the real Internet end-to-end delay values from King's dataset [15] to model the overlay node-to-node delay. Both the IP and overlay layers use shortest-path routing protocol. The overlay nodes deploy a RON-like overlay probing protocol for link monitoring and failure detection. To perform failure recovery and re-routing, the overlay nodes use an OSPF-like link state routing protocol.

We focus on overlay networks with fixed number of nodes, randomly chosen but fixed node locations, flexible overlay topology and timer mechanisms. This is due to the practical consideration that overlay administrators, e.g., PlanetLab, often rely on voluntarily donated servers and cannot determine the number of nodes and their placements. They can however, at their discretion, form the desired overlay topology and set their timer values. It is still possible for administrator to choose the number of overlay nodes and their placements if they have a large pool of candidates. However, this issue is not the focus of this paper.

4.1.2. Failure model

Failures are uniformly distributed across different IP links. The starting time of each IP-link failure is uniformly distributed over the simulation time. Since multiple IP paths (overlay links) might share the same failed IP link, one IP failure might trigger multiple overlay link failures. Note that even for overlay link failures triggered by the same IP failure events, the end-to-end observed failure duration can be very different. Unless otherwise specified, the IP-path failure duration (or TTR) used in our simulation has a distribution as displayed in Eq. (20).

$$P(t)=\begin{cases}0.067\times \mathrm{e}^{-0.0289\times t} & \text{if } t\leqslant 31.95,\\ 1-19t^{-0.85} & \text{if } t>31.95.\end{cases}\tag{20}$$

The following description explains how we determined this model. Based on large scale connectivity traces, Dahlin et al. [16] observed that 30% of IP path unavailabilities have duration longer than 30 s, and their failure duration is well modeled as $P'(t)=1-19t^{-0.85}$. The duration distribution of failures lasting less than 30 s is not presented in [16]. Assuming this portion of failure durations are exponentially distributed with mean λ_x, and based on the constraint of their ratio (Eq. (21)) and assuming a continuous boundary condition (Eq. (22))

$$\int_0^{30}\frac{\alpha}{\lambda_x}\mathrm{e}^{-\frac{t}{\lambda_x}}\mathrm{d}t=0.70,\tag{21}$$

$$1-19t^{-0.85}|_{t=31.95+}=\frac{\alpha}{\lambda_x}\mathrm{e}^{-\frac{t}{\lambda_x}}|_{t=31.95-},\tag{22}$$

we get $\alpha=2.32$ and $\lambda_x=34.6$.

This distribution is heavy-tailed and has a mean of infinity due to the $t>31.95$ portion [16]. To avoid being biased by a few outliers with large failure duration, we cut off the distribution at $t=100$. In addition to this realistic WAN failure model, we also simulate synthetic failure durations with exponential distribution of mean λ.

4.2. Failure detection

An overlay network must be able to detect an IP-layer link failure before it can provide any recovery service. Therefore, under a given constraint of overhead, a good overlay design should detect as many failures as possible.

In Fig. 9(a) and (b), we study the effect of probing timer value when using a full-mesh topology for failure detection. Failure detection ratio decreases linearly with an increasing T_{p}. Assuming an exponentially distributed failure duration with mean of 60 s, the default timer setting of RON will allow us to detect about 80% of the failure events. For the realistic WAN failure model, we can detect about 60% of the failure events. Fig. 9(b) looks at failure detection from the perspective of *Time-to-Detect* (TTD). Overlay probing mechanism is able to detect failures with

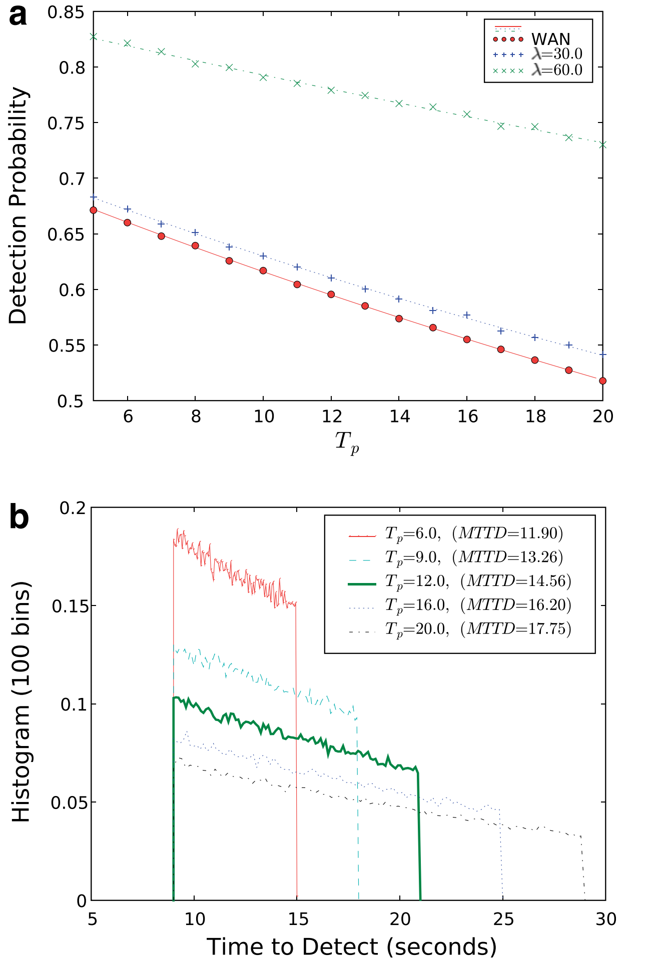


Fig. 9. Impact of setting T_{p}. (a) T_{p} vs. detection ratio and (b) T_{p} vs. TTD.

a TTD ranging from kT_t to $T_{\mathrm{p}}+kT_t$. Using a smaller T_{p} can reduce the range of the TTD distribution.

In Fig. 10, we study the failure detection ratio under different combinations of node degree (D) and T_{p} value. It can be observed in Fig. 10 that a smaller T_{p} or a larger connection degree (D) can detect more failures. However, as discussed in Section 3.3, the overhead is proportional to D/T_{p}. Therefore, given a constraint on overhead, one can choose the value of D or T_{p} to optimize the detection ratio. In the following, we will use $L=D/T_{\mathrm{p}}$ to measure the level of probing overhead. The default setting, full-mesh, $T_{\mathrm{p}}=12$ s in a 50-node overlay has $L=4.083$.

One can notice from Fig. 10 that the detection ratio has a steeper curve vs. connection degree D in contrast to T_{p}. Therefore, a larger T_{p} and D combination is the optimized value. This can be confirmed in Fig. 11. At different levels of overhead, a larger T_{p} (and D) results in higher failure detection ratio. Therefore, for a maximum failure detection ratio, an overlay administrator should build a

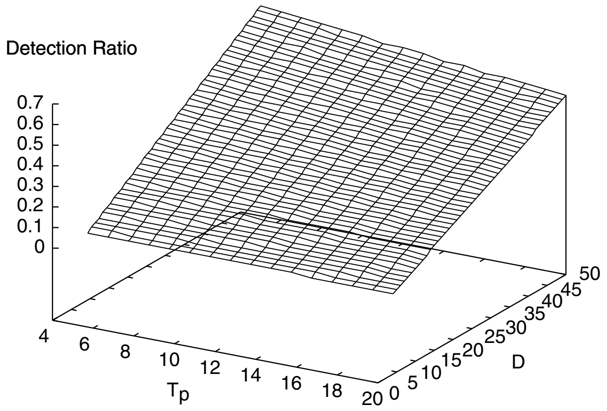


Fig. 10. T_p and D on detection ratio.

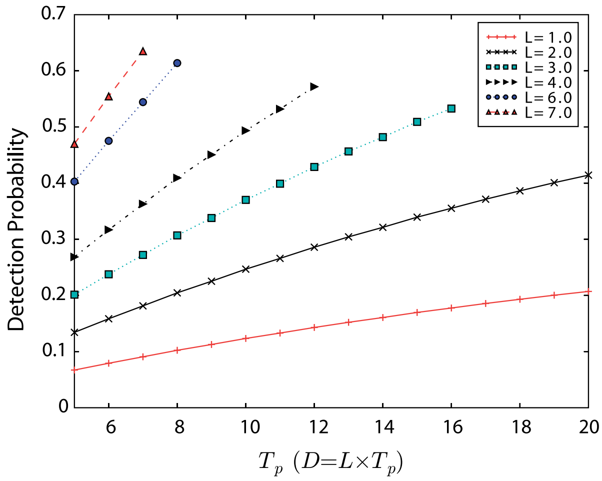


Fig. 11. Probing overhead.

full-mesh (largest D) and set the T_p large enough (moving towards the right) so that it satisfies the overhead constraint. When acceptable overhead is large, one can afford to have a smaller T_p and larger D (moving towards the top left corner) for larger failure detection ratio.

4.3. Failure recovery

Once an overlay monitoring session detects a failure, it will announce the failure events to its neighbors through Overlay-layer Link-State Advertisements (OLSAs). In the mean time, it will perform failure recovery by looking for an alternative route. As discussed in the previous section, it is best for the overlay network to monitor overlay links based on topologies with higher node degree for failure detection purpose. However, the end goal of overlay network is not only to detect failures but also to provide failure recovery services.

As presented in Fig. 6, increasing node degree providing negligible improvement on the ideal failure recovery ratio when $D \geqslant 4$, especially if failures involve only one IP link. In Section 3.6, we discussed that additional overlay links may actually delay network convergence, since overlay networks tend to observe multiple correlated link failures. Consequently, a topology with higher node degree may not be the better choice for failure recovery purposes, which, our simulation results below confirms.

In the following part, we call the full-mesh topology used for failure detection as the *probing graph*. We use a subgraph of the probing graph, called *recovery graph* for failure recovery.[1] The recovery graph contains all the overlay nodes with a smaller average node degree (D_R). Based on these two graphs, OLSA will be propagated only if a detected overlay link failure also affects recovery graph. However, for each detected failure, overlay nodes always try to re-route the failure (finding alternate overlay paths) through the recovery graph.

Fig. 12(a) depicts the effect of T_p on TTR based on a recovery graph of node degree 4. Most failures are recovered within $\langle kT_t, T_p + kT_t\rangle$. This is similar to T_p's effect on TTD. However, a careful comparison of Fig. 12(a) and 9(b) reveals the differences of slope. The distributions of TTD tend to be higher near the left size while the distributions of TTR tend to be higher near the right side. This is due to the fact that overlay network cannot always recover from a failure immediately upon detection. Instead, successful recovery is often delayed until each of the correlated overlay failures is detected and each corresponding OLSA is propagated.

Fig. 13(a) shows that increasing T_p will increase MTTR, or decrease AFDR. This is because a larger T_p not only reduces the failure detection ratio but also delays the start of the failure recovery process. On the positive side, a larger T_p value makes the overlay network less responsive to short-lived failures. By not responding to short-lived failures, the harmful effect on AFDR is minimum and network is more stable (as shown in Fig. 14(a)).

Figs. 12(b), 13(b) and 14(b) present the impact of introducing a hold-off timer. Hold-off timer essentially delays the recovery process for failures detected earlier, accumulates failures (OLSAs) over T_{rt} time, and makes one combined route recalculation. A larger T_{rt} will allow the overlay node to accumulate more correlated OLSAs and reduces re-calculation. This can effectively improve network

[1] The detail motivation of separating these two graphs will be expanded in the following paragraphs.

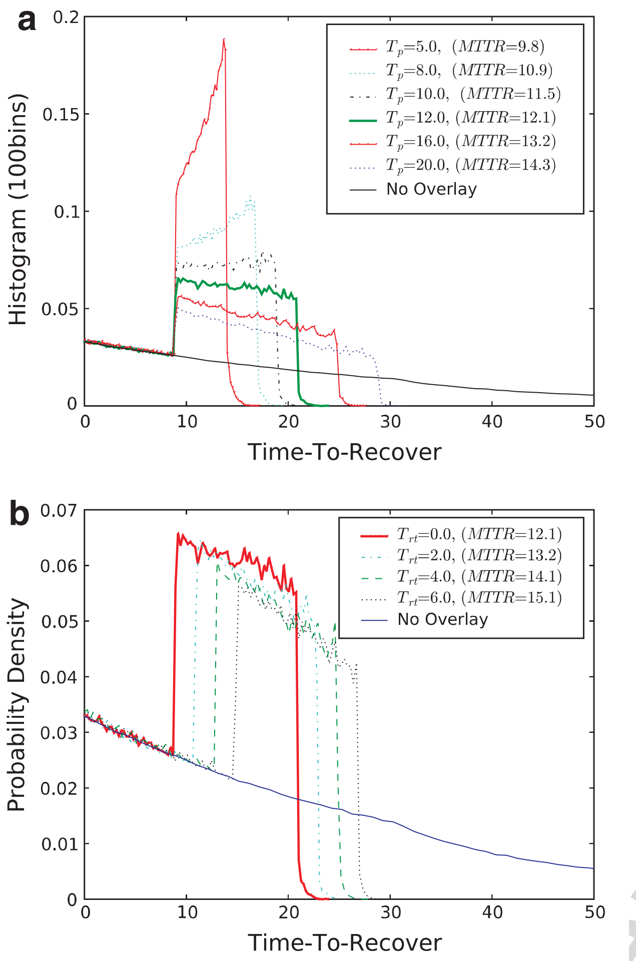


Fig. 12. Timer on TTR distribution. (a) T_{rt} vs. TTR and (b) T_{rt} vs. TTR.

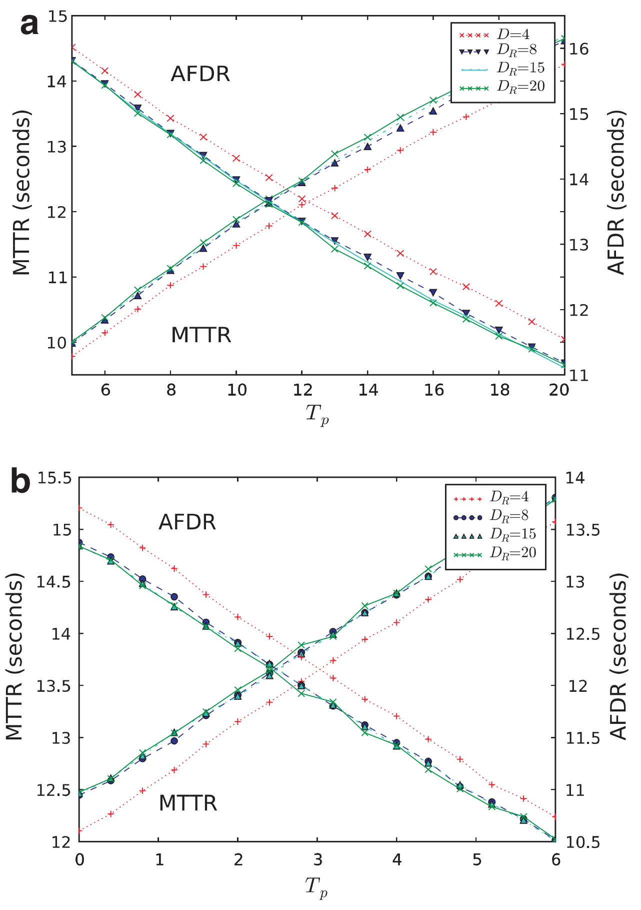


Fig. 13. Timer on recovery performance. (a) $T_{\rm p}$ vs. AFDR and (b) T_{rt} vs. AFDR.

stability, as can be observed in Fig. 14(a). On the negative side, introducing a hold-off timer clearly moves the distribution of TTR to the right and increases the TTR (Fig. 12(b)).

In Fig. 13(a), we note that using a recovery graph with smaller node degree actually results in smaller MTTR (or larger AFDR). This is because an increase in node degree helps little in ideal recovery ratio but increases the chance of delayed re-convergence. In addition, Fig. 14(a) shows that a smaller node degree incurs fewer path changes, thus improving network stability. Similar results can be observed from Figs. 13(b) and 14(b). Therefore, increasing the node degree is counterproductive for failure recovery.

4.4. Suggestions and discussions

Based on the analysis and simulation results, we believe the following techniques can be adopted to design overlay networks with greater performance.

- The node degree (D) have opposite impacts on failure detection and failure recovery. For failure detection, a larger node degree is preferred. In contrast, a smaller node degree is preferred for failure recovery. Using a full-mesh graph for failure detection and a graph with smaller node degree for failure recovery can exploit the best of both approaches. Practically, we believe that little modification is needed to achieve this goal. In addition, end nodes/applications may take the responsibility of detecting path failures while the overlay networks are only in charge of providing resilient end-to-end forwarding paths.
- The timer values present a fundamental tradeoff between network stability, overhead, and failure recovery performance. The probing timer ($T_{\rm p}$) can improve failure detection and recovery at the price of probing overhead and stability. Using a hold-off timer (T_{rt}), which is not currently implemented in popular overlay networks,

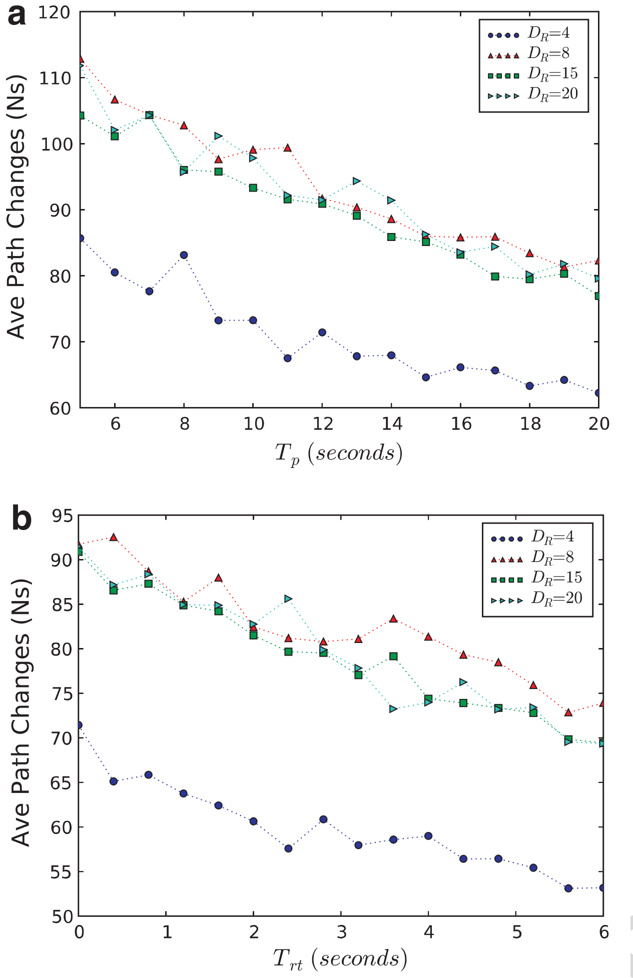


Fig. 14. Timer on network stability. (a) T_p vs. N_s and (b) T_{rt} vs. N_s.

can improve the network stability while minimally affect failure recovery performance.

- Several people have proposed to build overlay network with topology-awareness [17,18]. This could fundamentally change the scenario of overlay failure detection and recovery. With the knowledge of lower-layer topology, one could make meaningful inference about the status of one link from the status of other links. If topology-aware overlay becomes a reality, the failure detection algorithms based on information sharing [11] could speed up overlay failure detection and recovery significantly. Topology-awareness and information sharing can also relieve the overlay from doing an O($n \times k$) (n is number of overlay nodes and k is average number of peers) probing and significantly reduce the overhead. We plan to investigate this approach thoroughly in the future work.

5. Related work

Zhuang et al. [11] investigate the tradeoffs of different overlay/P2P *node* failure detection algorithms in terms of overhead, packet loss ratio and failure detection ratio. In addition, the paper also recommends several mechanisms to design optimal node failure detection methods. The same topic is also discussed in [19], in which the authors focus on analytical models and propose a self-tuning method. Instead of node failure detection, we focus on the issue of overlay *link* failure detection and recovery in the paper. Moreover, we focus on the impact of IP-layer failure characteristics on probing interval setup and explore the tradeoff between failure recovery performance and overhead.

Some work has been done on setting up optimal hello message intervals in OSPF network environment. Goyal et al. [20] investigate the impact of topologies and network congestion on optimal `HelloInterval` for OSPF network through simulation. Basu et al. [21] perform experimental study of the stability of OSPF in terms of convergence time, routing load and number of routing flaps. In [22], the authors use analytical methods to study the effects of traffic overload on OSPF and BGP by quantifying the stability and robustness properties.

Qiu et al. [23] studied the *vertical interaction* between selfish overlay network and lower-layer traffic engineering mechanisms. Based on the network traffic pattern, traffic engineering mechanism, e.g., OSPF and MPLS optimization, aims to achieve optimal network performance by adjusting routing at its respective layer. On the other hand, overlay nodes aim to find the best route for themselves and can affect the traffic demands observed by lower-layer. Qiu et al. show that the interplay of selfish overlay and traffic engineering can result in a system performance worse than using either one of them. Keralapura et al. [24] discuss the possible interaction between the IP layer and overlay networks that may affect traffic matrix estimation and load balancing, or lead to oscillatory race conditions between different overlay networks. Liu et al. [25] model the interaction between an overlay network and traffic engineering work as a two-players game. Our work is complementary to the prior studies and provides new insights on using overlay to provide failure recovery services to IP networks.

6. Conclusion

In this paper, we perform analysis and simulation studies to investigate failure detection and recovery process in overlay networks. In particular, we study how the different parameters (T_p, T_{rt} and D) impact the performance of the overlay. These parameters are important factors in the tradeoffs between performance, probing overhead, and penalty (e.g., oscillation of user perceived performance). A well-designed overlay network should set the parameters carefully to meet its goals and constraints. Since node degree has opposite effects on failure detection and recovery, we made a novel proposal that one should use a topology with high node degree for failure detection and another topology with small node degree for failure recovery. Our analysis and results provide important guidelines to design overlay networks and to understand their performance.

References

[1] A. Markopoulou, G. Iannaconne, S. Bhattacharrya, C.N. Chuah, C. Diot, Characterization of failures in an IP backbone, in: Proc. IEEE INFOCOM, 2004, pp. 2307–2317.

[2] R. Mahajan, D. Wetherall, T. Anderson, Understanding BGP Misconfiguration, in: Proc. ACM SIGCOMM, 2002, pp. 3–17.

[3] V.E. Paxson, Measurements and analysis of end-to-end Internet dynamics, Ph.D. thesis, UC Berkeley (April 1997).

[4] K.P. Gummadi, H.V. Madhyastha, S.D. Gribble, H.M. Levy, D. Wetherall, Improving the reliability of internet paths with one-hop source routing, in: Proc. USENIX/ACM Symposium on Operating Systems Design and Implementation, 2004, pp. 183–198.

[5] D.G. Andersen, H. Balakrishnan, M.F. Kaashoek, R. Morris, Resilient overlay network, in: Proc. ACM Symposium on Operating Systems Principles, 2001, pp. 131–145.

[6] Akamai Corporation, http://www.akamai.com.

[7] S. Ratnasamy, P. Francis, M. Handley, R. Karp, S. Shenker, A scalable content-addressable network, in: Proc. ACM SIGCOMM, 2001, pp. 161–172.

[8] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, H. Balakrishnan, Chord: A scalable peer-to-peer lookup service for internet applications, in: Proc. ACM SIGCOMM, 2001, 149–160.

[9] Gnutella, http://www.gnutella.com.

[10] Planet-lab Network Testbed, http://www.planet-lab.org.

[11] S.Q. Zhuang, D. Geels, I. Stoica, R.H. Katz, On failure detection algorithms in overlay networks, in: Proc. IEEE INFOCOM, 2005, pp. 2112–2123.

[12] S.Banerjee, S. Jee, B.Bhattacharjee, C.Kommareddy, Resilient multicast using overlays, in: Proc. ACM SIGMETRICS, 2003, pp. 102–113.

[13] Y. Amir, C. Danilov, S. Goose, D. Hedqvist, A. Terzis, An overlay architecture for high quality VoIP streams, IEEE Transactions on Multimedia, 2006.

[14] A. Medina, A. Lakhina, I. Matta, J. Byers, BRITE, http://www.cs.bu.edu/brite/, 2002.

[15] K.P. Gummadi, S. Saroiu, S.D. Gribble, King: Estimating latency between arbitrary internet end hosts, in: Proc. Internet Measurement Workshop, 2002, pp. 5–18.

[16] M. Dahlin, B.B.V. Chandra, L. Gao, A. Nayate, End-to-end WAN service availability, IEEE/ACM Trans. Networking.

[17] S. Ratnasamy, M. Handley, R. Karp, S. Shenker, Topologically-aware overlay construction and server selection, in: Proc. IEEE INFOCOM, 2002, pp. 1190–1199.

[18] J. Han, D. Watson, F. Jahanian, Topology aware overlay networks, in: Proc. IEEE INFOCOM, 2005, pp. 2554–2565.

[19] R. Mahajan, M. Castro, A. Rowstron, Controlling the cost of reliability in peer-to-peer overlays, in: Proc. International Workshop on Peer-to-Peer Systems, 2003, pp. 21–32.

[20] M. Goyal, K.K. Ramakrishnan, W. Feng, Achieving faster failure detection in OSPF networks, in: Proc. International Conference on Communications, 2003, pp. 296–300.

[21] A. Basu, J.G. Riecke, Stability issues in OSPF routing, in: Proc. ACM SIGCOMM, 2002, pp. 225–236.

[22] A. Shaikh, L. Kalampoukas, R. Dube, A. Varma, Routing stability in congested networks: Experimentation and analysis, in: Proc. ACM SIGCOMM, 2000, pp. 163–174.

[23] L. Qiu, Y.R. Yang, Y. Zhang, S. Shenker, On selfish routing in internet-like environments, in: Proc. ACM SIGCOMM, 2003, pp. 151–162.

[24] R. Keralapura, N. Taft, C.-N. Chuah, G. Iannaccone, Can ISPs take the heat from overlay networks?, in: Proc. ACM HotNets, 2004, pp. 27–29.

[25] Y. Liu, H. Zhang, W. Gong, D. Towsley, On the interaction between overlay routing and traffic engineering, in: Proc. IEEE INFOCOM, 2005, pp. 2543–2553.

Zhi Li is currently a Senior Member of Technical Staff at AT&T labs, AT&T Services. He received his Ph.D. (2005) and Masters (2003) in Computer Science, University of California at Davis, Masters (2000) in Computer Engineering at Tsinghua University, China. His research interests include video over IP, Quality-of-Service, Multicasting, overlay networks, traffic engineering and Modeling.

Lihua Yuan is currently a Ph.D Candidate in the Department of Electrical and Computer Engineering at the University of California, Davis. He received his Bachelor's degree in Electrical and Electronics Engineering from Nanyang Technological University (Singapore) and Master's degree in Electrical and Computer Engineering from National University of Singaore (Singapore). His research interests are in the area of network management, management and security.

Prasant Mohapatra is currently a Professor in the Department of Computer Science at the University of California, Davis. In the past, he was on the faculty at Iowa State University and Michigan State University. He has also held Visiting Scientist positions at Intel Corporation, Panasonic Technologies, Institute of Infocomm Research (I2R), Singapore, and National ICT Australia (NICTA). He received his Ph.D. in Computer Engineering from the Pennsylvania State University in 1993. He was/is on the editorial board of the IEEE Transactions on computers, IEEE Transaction on Parallel and Distributed Systems, ACM WINET, and Ad Hoc Networks. He has been on the program/organizational committees of several international conferences. He was the Program Vice-Chair of INFOCOM 2004, and the Program Co-Chair of the First IEEE International Conference on Sensor and Ad Hoc Communications and Networks (SECON 2004). He has been a Guest Editor for IEEE Network, IEEE Transactions on Mobile Computing, and the IEEE Computer.

His research interests are in the areas of wireless networks, sensor networks, Internet protocols and QoS. His research has been funded though grants from the National Science Foundation, Intel Corporation, Siemens, Panasonic Technologies, Hewlett Packard, and EMC Corporation.

Chen-Nee Chuah (SM'06) is currently an Associate Professor in the Electrical and Computer Engineering Department at the University of California, Davis (UCD). She received her B.S. in Electrical Engineering from Rutgers University in 1995, and her M.S. and Ph.D. in Electrical Engineering and Computer Sciences from the University of California, Berkeley in 1997 and 2001, respectively. Before joining UCD, she held a visiting researcher position at Sprint Advanced Technology Laboratories. Her research interests are in the area of computer networking and distributed systems, Internet measurements, overlay/peer-to-peer systems, network security, wireless/mobile networking, and opportunistic communications. She received the National Science Foundation CAREER Award in 2003 and the UC Davis College of Engineering Outstanding Junior Faculty Award in 2004. She has served as the Technical Program Co-Chair for the ACM Mobicom 2004 Workshop on Vehicular Ad Hoc Networks (VANET), and the Vice-Chair for IEEE Globecom 2006. She has also served on the technical program committee of several ACM- and IEEE-sponsored conferences.